The Best Lexical Metric for Phrase-Based Statistical MT System Optimization

Daniel Cer, Christopher D. Manning and Daniel Jurafsky
Stanford University
Stanford, CA 94305, USA

Abstract

Translation systems are generally trained to optimize BLEU, but many alternative metrics are available. We explore how optimizing toward various automatic evaluation metrics (BLEU, METEOR, NIST, TER) affects the resulting model. We train a state-of-the-art MT system using MERT on many parameterizations of each metric and evaluate the resulting models on the other metrics and also using human judges. In accordance with popular wisdom, we find that it's important to train on the same metric used in testing. However, we also find that training to a newer metric is only useful to the extent that the MT model's structure and features allow it to take advantage of the metric. Contrasting with TER's good correlation with human judgments, we show that people tend to prefer BLEU and NIST trained models to those trained on edit distance based metrics like TER or WER. Human preferences for METEOR trained models varies depending on the source language. Since using BLEU or NIST produces models that are more robust to evaluation by other metrics and perform well in human judgments, we conclude they are still the best choice for training.

1 Introduction

Since their introduction, automated measures of machine translation quality have played a critical role in the development and evolution of SMT systems. While such metrics were initially intended for evaluation, popular training methods such as minimum error rate training (MERT) (Och, 2003) and margin infused relaxed algorithm (MIRA) (Crammer and Singer, 2003; Watanabe et al., 2007; Chiang et al., 2008) train translation models toward a specific evaluation metric. This makes the quality of the resulting model dependent on how accurately the automatic metric actually reflects human preferences.

The most popular metric for both comparing systems and tuning MT models has been BLEU. While BLEU (Papineni et al., 2002) is relatively simple, scoring translations according to their n-gram overlap with reference translations, it still achieves a reasonable correlation with human judgments of translation quality. It is also robust enough to use for automatic optimization. However, BLEU does have a number of shortcomings. It doesn't penalize n-gram scrambling (Callison-Burch et al., 2006), and since it isn't aware of synonymous words or phrases, it can inappropriately penalize translations that use them.

Recently, there have been efforts to develop better evaluation metrics. Metrics such as *Translation Edit Rate* (TER) (Snover et al., 2006; Snover et al., 2009) and METEOR[1] (Lavie and Denkowski, 2009) perform a more sophisticated analysis of the translations being evaluated and the scores they produce tend to achieve a better correlation with human judgments than those produced by BLEU (Snover et al., 2009; Lavie and Denkowski, 2009; Przybocki et al., 2008; Snover et al., 2006).

Their better correlations suggest that we might obtain higher quality translations by making use of these new metrics when training our models. We expect that training on a specific metric will produce the best performing model according to that met-

[1]METEOR: Metric for Evaluation of Translation with Explicit ORdering.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 555–563,
Los Angeles, California, June 2010.

ric. Doing better on metrics that better reflect human judgments seems to imply the translations produced by the model would be preferred by human judges.

However, there are four potential problems. First, some metrics could be susceptible to systematic exploitation by the training algorithm and result in model translations that have a high score according to the evaluation metric but that are of low quality.[2] Second, other metrics may result in objective functions that are harder to optimize. Third, some may result in better generalization performance at test time by not encouraging overfitting of the training data. Finally, as a practical concern, metrics used for training cannot be too slow.

In this paper, we systematically explore these four issues for the most popular metrics available to the MT community. We examine how well models perform both on the metrics on which they were trained and on the other alternative metrics. Multiple models are trained using each metric in order to determine the stability of the resulting models. Select models are scored by human judges in order to determine how performance differences obtained by tuning to different automated metrics relates to actual human preferences.

The next sections introduce the metrics and our training procedure. We follow with two sets of core results, machine evaluation in section 5, and human evaluation in section 6.

2 Evaluation Metrics

Designing good automated metrics for evaluating machine translations is challenging due to the variety of acceptable translations for each foreign sentence. Popular metrics produce scores primarily based on matching sequences of words in the system translation to those in one or more reference translations. The metrics primarily differ in how they account for reorderings and synonyms.

2.1 BLEU

BLEU (Papineni et al., 2002) uses the percentage of n-grams found in machine translations that also occur in the reference translations. These n-gram precisions are calculated separately for different n-gram lengths and then combined using a geometric mean. The score is then scaled by a brevity penalty if the candidate translations are shorter than the references, $BP = \min(1.0, e^{1-len(R)/len(T)})$. Equation 1 gives BLEU using n-grams up to length N for a corpus of candidate translations T and reference translations R. A variant of BLEU called the NIST metric (Doddington, 2002) weights n-gram matches by how informative they are.

$$\text{BLEU:}N = \left(\prod_{n=1}^{N} \frac{n\text{-grams}(T \bigcap R)}{n\text{-grams}(T)} \right)^{\frac{1}{N}} BP \quad (1)$$

While easy to compute, BLEU has a number of shortcomings. Since the order of matching n-grams is ignored, n-grams in a translation can be randomly rearranged around non-matching material or other n-gram breaks without harming the score. BLEU also does not explicitly check whether information is missing from the candidate translations, as it only examines what fraction of candidate translation n-grams are in the references and not what fraction of references n-grams are in the candidates (i.e., BLEU ignores n-gram recall). Finally, the metric does not account for words and phrases that have similar meanings.

2.2 METEOR

METEOR (Lavie and Denkowski, 2009) computes a one-to-one alignment between matching words in a candidate translation and a reference. If a word matches multiple other words, preference is given to the alignment that reorders the words the least, with the amount of reordering measured by the number of crossing alignments. Alignments are first generated for exact matches between words. Additional alignments are created by repeatedly running the alignment procedure over unaligned words, first allowing for matches between word stems, and then allowing matches between words listed as synonyms in WordNet. From the final alignment, the candidate translation's unigram precision and recall is calculated, $P = \frac{\text{matches}}{\text{length trans}}$ and $R = \frac{\text{matches}}{\text{length ref}}$. These two are then combined into a weighted harmonic mean (2). To penalize reorderings, this value is then scaled by a fragmentation penalty based on the number of chunks the two sentences would need to be broken

[2] For example, BLEU computed without the brevity penalty would likely result in models that have a strong preference for generating pathologically short translations.

into to allow them to be rearranged with no crossing alignments, $P_{\beta,\gamma} = 1 - \gamma\left(\frac{\text{chunks}}{\text{matches}}\right)^{\beta}$.

$$F_\alpha = \frac{PR}{\alpha P + (1-\alpha)R} \tag{2}$$

$$\text{METEOR}_{\alpha,\beta,\gamma} = F_\alpha \cdot P_{\beta,\gamma} \tag{3}$$

Equation 3 gives the final METEOR score as the product of the unigram harmonic mean, F_α, and the fragmentation penalty, $P_{\beta,\gamma}$. The free parameters α, β, and γ can be used to tune the metric to human judgments on a specific language and variation of the evaluation task (e.g., ranking candidate translations vs. reproducing judgments of translations adequacy and fluency).

2.3 Translation Edit Rate

TER (Snover et al., 2006) searches for the shortest sequence of edit operations needed to turn a candidate translation into one of the reference translations. The allowable edits are the insertion, deletion, and substitution of individual words and swaps of adjacent sequences of words. The swap operation differentiates TER from the simpler word error rate (WER) metric (Nießen et al., 2000), which only makes use of insertions, deletions, and substitutions. Swaps prevent phrase reorderings from being excessively penalized. Once the shortest sequence of operations is found,[3] TER is calculated simply as the number of required edits divided by the reference translation length, or average reference translation length when multiple are available (4).

$$\text{TER} = \frac{\min \text{edits}}{\text{avg ref length}} \tag{4}$$

TER-Plus (TERp) (Snover et al., 2009) extends TER by allowing the cost of edit operations to be tuned in order to maximize the metric's agreement with human judgments. TERp also introduces three new edit opertions: word stem matches, WordNet synonym matches, and multiword matches using a table of scored paraphrases.

[3]Since swaps prevent TER from being calculated exactly using dynamic programming, a beam search is used and this can overestimate the number of required edits.

3 MERT

MERT is the standard technique for obtaining a machine translation model fit to a specific evaluation metric (Och, 2003). Learning such a model cannot be done using gradient methods since the value of the objective function only depends on the translation model's argmax for each sentence in the tuning set. Typically, this optimization is performed as a series of line searches that examines the value of the evaluation metric at critical points where a new translation argmax becomes preferred by the model. Since the model score assigned to each candidate translation varies linearly with changes to the model parameters, it is possible to efficiently find the global minimum along any given search direction with only $O(n^2)$ operations when n-best lists are used.

Using our implementation of MERT that allows for pluggable optimization metrics, we tune models to BLEU:N for $N = 1 \ldots 5$, TER, two configurations of TERp, WER, several configurations of METEOR, as well as additive combinations of these metrics. The TERp configurations include the default configuration of TERp and TERpA: the configuration of TERp that was trained to match human judgments for NIST Metrics MATR (Matthew Snover and Schwartz, 2008; Przybocki et al., 2008). For METEOR, we used the standard METEOR English parameters ($\alpha = 0.8, \beta = 2.5, \gamma = 0.4$), and the English parameters for the ranking METEOR ($\alpha = 0.95, \beta = 0.5, \gamma = 0.5$),[4] which was tuned to maximize the metric's correlation with WMT-07 human ranking judgements (Agarwal and Lavie, 2008). The default METEOR parameters favor longer translations than the other metrics, since high α values place much more weight on unigram recall than precision. Since this may put models tuned to METEOR at a disadvantage when being evaluated by the other metrics, we also use a variant of the standard English model and of ranking METEOR with α set to 0.5, as this weights both recall and precision equally.

For each iteration of MERT, 20 random restarts were used in addition to the best performing point discovered during earlier iterations of training.[5]

[4]Agarwal and Lavie (2008) report $\gamma = 0.45$, however the 0.8.2 release of METEOR uses $\gamma = 0.5$ for ranking English.

[5]This is not necessarily identical with the point returned by the most recent MERT iteration, but rather can be any point

Since MERT is known to be sensitive to what restart points are provided, we use the same series of random restart points for each model. During each iteration of MERT, the random seed is based on the MERT iteration number. Thus, while a different set of random points is selected during each MERT iteration, on any given iteration all models use the same set of points. This prevents models from doing better or worse just because they received different starting points. However, it is still possible that certain random starting points are better for some evaluation metrics than others.

4 Experiments

Experiments were run using Phrasal (Cer et al., 2010), a left-to-right beam search decoder that achieves a matching BLEU score to Moses (Koehn et al., 2007) on a variety of data sets. During decoding we made use of a stack size of 100, set the distortion limit to 6, and retrieved 20 translation options for each unique source phrase.

Using the selected metrics, we train both Chinese to English and Arabic to English models.[6] The Chinese to English models are trained using NIST MT02 and evaluated on NIST MT03. The Arabic to English experiments use NIST MT06 for training and GALE dev07 for evaluation. The resulting models are scored using all of the standalone metrics used during training.

4.1 Arabic to English

Our Arabic to English system was based on a well ranking 2009 NIST submission (Galley et al., 2009). The phrase table was extracted using all of the allowed resources for the constrained Arabic to English track. Word alignment was performed using the Berkeley cross-EM aligner (Liang et al., 2006). Phrases were extracted using the grow heuristic (Koehn et al., 2003). However, we threw away all phrases that have a $P(e|f) < 0.0001$ in order to reduce the size of the phrase table. From the aligned data, we also extracted a hierarchical reordering model that is similar to popular lexical reordering models (Koehn et al., 2007) but that models swaps containing more than just one phrase (Galley and Manning, 2008). A 5-gram language model was created with the SRI language modeling toolkit (Stolcke, 2002) using all of the English material from the parallel data employed to train the phrase table as well as Xinhua Chinese English Parallel News (LDC2002E18).[7] The resulting decoding model has 16 features that are optimized during MERT.

4.2 Chinese to English

For our Chinese to English system, our phrase table was built using 1,140,693 sentence pairs sampled from the GALE Y2 training data. The Chinese sentences were word segmented using the 2008 version of Stanford Chinese Word Segmenter (Chang et al., 2008; Tseng et al., 2005). Phrases were extracted by running GIZA++ (Och and Ney, 2003) in both directions and then merging the alignments using the grow-diag-final heuristic (Koehn et al., 2003). From the merged alignments we also extracted a bidirectional lexical reordering model conditioned on the source and the target phrases (Koehn et al., 2007). A 5-gram language model was created with the SRI language modeling toolkit (Stolcke, 2002) and trained using the Gigaword corpus and English sentences from the parallel data. The resulting decoding model has 14 features to be trained.

5 Results

As seen in tables 1 and 2, the evaluation metric we use during training has a substantial impact on performance as measured by the various other metrics. There is a clear block structure where the best class of metrics to train on is the same class that is used during evaluation. Within this block structure, we make three primary observations. First, the best performing model according to any specific metric *configuration* is usually not the model we trained to that configuration. In the Chinese results, the model trained on BLEU:3 scores 0.74 points higher on BLEU:4 than the model actually trained to BLEU:4. In fact, the BLEU:3 trained model outperforms all other models on BLEU:N metrics. For the Arabic results, training on NIST scores 0.27 points higher

returned during an earlier iteration of MERT.

[6] Given the amount of time required to train a TERpA model, we only present TERpA results for Chinese to English.

[7] In order to run multiple experiments in parallel on the computers available to us, the system we use for this work differs from our NIST submission in that we remove the Google n-gram language model. This results in a performance drop of less than 1.0 BLEU point on our dev data.

Train\Eval	BLEU:1	BLEU:2	BLEU:3	BLEU:4	BLEU:5	NIST	TER	TERp	WER	TERpA	METR	METR-r	METR $\alpha = 0.5$	METR-r $\alpha = 0.5$
BLEU:1	75.98	55.39	40.41	29.64	21.60	11.94	78.07	78.71	68.28	73.63	41.98	59.63	42.46	60.02
BLEU:2	76.58	57.24	42.84	32.21	24.09	12.20	77.09	77.63	67.16	72.54	43.20	60.91	43.59	61.56
BLEU:3	**76.74**	**57.46**	**43.13**	**32.52**	**24.44**	12.22	76.53	77.07	66.81	72.01	42.94	60.57	43.40	60.88
BLEU:4	76.24	56.86	42.43	31.80	23.77	12.14	76.75	77.25	66.78	72.01	43.29	60.94	43.10	61.27
BLEU:5	76.39	57.14	42.93	32.38	24.33	12.40	75.42	75.77	65.86	70.29	43.02	61.22	43.57	61.43
NIST	76.41	56.86	42.34	31.67	23.57	12.38	75.20	75.72	65.78	70.11	43.11	61.04	43.78	**61.84**
TER	73.23	53.39	39.09	28.81	21.18	**12.73**	**71.33**	**71.70**	63.92	**66.58**	38.65	55.49	41.76	59.07
TERp	72.78	52.90	38.57	28.32	20.76	12.68	71.76	72.16	64.26	66.96	38.51	56.13	41.48	58.73
TERpA	71.79	51.58	37.36	27.23	19.80	12.54	72.26	72.56	64.58	67.30	37.86	55.10	41.16	58.04
WER	74.49	54.59	40.30	29.88	22.14	12.64	71.85	72.34	**63.82**	67.11	39.76	57.29	42.37	59.97
METR	73.33	54.35	40.28	30.04	22.39	11.53	84.74	85.30	71.49	79.47	**44.68**	62.14	42.99	60.73
METR-r	74.20	54.99	40.91	30.66	22.98	11.74	82.69	83.23	70.49	77.77	44.64	**62.25**	43.44	61.32
METR:0.5	76.36	56.75	42.48	31.98	24.00	12.44	74.94	75.32	66.09	70.14	42.75	60.98	**43.86**	61.38
METR-r:0.5	76.49	56.93	42.36	31.70	23.68	12.21	77.04	77.58	67.12	72.23	43.26	61.03	43.63	61.67
Combined Models														
BLEU:4-TER	75.32	55.98	41.87	31.42	23.50	12.62	72.97	73.38	64.46	67.95	41.50	59.11	43.50	60.82
BLEU:4-2TERp	75.22	55.76	41.57	31.11	23.25	12.64	72.48	72.89	64.17	67.43	41.12	58.82	42.73	60.86
BLEU:4+2MTR	75.77	56.45	42.04	31.47	23.48	11.98	79.96	80.65	68.85	74.84	44.06	61.78	43.70	61.48

Table 1: Chinese to English test set performance on MT03 using models trained using MERT on MT02. In each column, cells shaded blue are better than average and those shaded red are below average. The intensity of the shading indicates the degree of deviation from average. For BLEU, NIST, and METEOR, higher is better. For edit distance metrics like TER and WER, lower is better.

Train\Eval	BLEU:1	BLEU:2	BLEU:3	BLEU:4	BLEU:5	NIST	TER	TERp	WER	METR	METR-r	METR $\alpha = 0.5$	METR-r $\alpha = 0.5$
BLEU:1	79.90	65.35	54.08	45.14	37.81	10.68	46.19	61.04	49.98	49.74	67.79	49.19	68.12
BLEU:2	80.03	65.84	54.70	45.80	38.47	10.75	45.74	60.63	49.24	50.02	68.00	49.71	68.27
BLEU:3	79.87	65.71	54.59	45.67	38.34	10.72	45.86	60.80	49.18	49.87	68.32	49.61	67.67
BLEU:4	80.39	66.14	54.99	46.05	38.70	10.82	45.25	59.83	48.69	49.65	68.13	49.66	67.92
BLEU:5	79.97	65.77	54.64	45.76	38.44	10.75	45.66	60.55	49.11	49.89	68.33	49.64	68.19
NIST	80.41	66.27	55.22	46.32	38.98	**10.96**	44.11	57.92	47.74	48.88	67.85	**49.88**	**68.52**
TER	79.69	65.52	54.44	45.55	38.23	10.75	43.36	56.12	**47.11**	47.90	66.49	49.55	68.12
TERp	79.27	65.11	54.13	45.35	38.12	10.75	**43.36**	**55.92**	47.14	47.83	66.34	49.43	67.94
WER	79.42	65.28	54.30	45.51	38.27	10.78	43.44	56.13	47.13	47.82	66.33	49.38	67.88
METR	75.52	60.94	49.84	41.17	34.12	9.93	52.81	70.08	55.72	50.92	68.55	48.47	66.89
METR-r	77.42	62.91	51.67	42.81	35.61	10.24	49.87	66.26	53.17	**50.95**	**69.29**	49.29	67.89
METR:0.5	79.69	65.14	53.94	45.03	37.72	10.72	45.80	60.44	49.34	49.78	68.31	49.23	67.72
METR-r:0.5	79.76	65.12	53.82	44.88	37.57	10.67	46.53	61.55	50.17	49.66	68.57	49.58	68.25
Combined Models													
BLEU:4-TER	80.37	66.31	55.27	46.36	39.00	10.96	43.94	57.46	47.46	49.00	67.10	49.85	68.41
BLEU:4-2TERp	79.65	65.53	54.54	45.75	38.48	10.80	43.42	56.16	47.15	47.90	65.93	49.09	67.90
BLEU:4+2METR	79.43	64.97	53.75	44.87	37.58	10.63	46.74	62.03	50.35	50.42	68.92	49.70	68.37

Table 2: Arabic to English test set performance on dev07 using models trained using MERT on MT06. As above, in each column, cells shaded blue are better than average and those shaded red are below average. The intensity of the shading indicates the degree of deviation from average.

on BLEU:4 than training on BLEU:4, and outperforms all other models on BLEU:N metrics.

Second, the edit distance based metrics (WER, TER, TERp, TERpA)[8] seem to be nearly interchangeable. While the introduction of swaps allows the scores produced by the TER metrics to achieve better correlation with human judgments, our models are apparently unable to exploit this during training. This maybe due to the monotone nature of the reference translations and the fact that having multiple references reduces the need for reorderings. However, it is possible that differences between training to WER and TER would become more apparent using models that allow for longer distance reorderings or that do a better job of capturing what reorderings are acceptable.

Third, with the exception of BLEU:1, the performance of the BLEU, NIST, and the METEOR α=.5 models appears to be more robust across the other evaluation metrics than the standard METEOR, METEOR ranking, and edit distance based models (WER, TER, TERp, an TERpA). The latter models tend to do quite well on metrics similar to what they were trained on, while performing particularly poorly on the other metrics. For example, on Chinese, the TER and WER models perform very well

[8] In our implementation of multi-reference WER, we use the length of the references that result in the lowest sentence level WER to divide the edit costs. In contrast, TER divides by the average reference length. This difference can sometimes result in WER being lower than the corresponding TER. Also, as can be seen in the Arabic to English results, TERp scores sometimes differ dramatically from TER scores due to normalization and tokenization differences (e.g., TERp removes punctuation prior to scoring, while TER does not).

on other edit distance based metrics, while performing poorly on all the other metrics except NIST. While less pronounced, the same trend is also seen in the Arabic data. Interestingly enough, while the TER, TERp and standard METEOR metrics achieve good correlations with human judgments, models trained to them are particularly mismatched in our results. The edit distance models do terribly on METEOR and METEOR ranking, while METEOR and METEOR ranking models do poorly on TER, TERp, and TERpA.

Training Metric	Itr	MERT Time	Training Metric	Itr	MERT Time
BLEU:1	13	21:57	NIST	15	78:15
BLEU:2	15	32:40	TER	7	21:00
BLEU:3	19	45:08	TERp	9	19:19
BLEU:4	10	24:13	TERpA	8	393:16
BLEU:5	16	46:12	WER	13	33:53
BL:4-TR	9	21:07	BL:4-2TRp	8	22:03
METR	12	39:16	METR 0.5	18	42:04
METR R	12	47:19	METR R:0.5	13	25:44

Table 3: Chinese to English MERT iterations and training times, given in hours:mins and excluding decoder time.

5.1 Other Results

On the training data, we see a similar block structure within the results, but there is a different pattern among the top performers. The tables are omitted, but we observe that, for Chinese, the BLEU:5 model performs best on the training data according to all higher order BLEU metrics (4-7). On Arabic, the BLEU:6 model does best on the same higher order BLEU metrics (4-7). By rewarding higher order n-gram matches, these objectives actually find minima that result in more 4-gram matches than the models optimized directly to BLEU:4. However, the fact that this performance advantage disappears on the evaluation data suggests these higher order models also promote overfitting.

Models trained on additive metric blends tend to smooth out performance differences between the classes of metrics they contain. As expected, weighting the metrics used in the additive blends results in models that perform slightly better on the type of metric with the highest weight.

Table 3 reports training times for select Chinese to English models. Training to TERpA is very computationally expensive due to the implementation of the paraphrase table. The TER family of metrics tends to converge in fewer MERT iterations than those trained on other metrics such as BLEU, METEOR or even WER. This suggests that the learning objective provided by these metrics is either easier to optimize or they more easily trap the search in local minima.

5.2 Model Variance

One potential problem with interpreting the results above is that learning with MERT is generally assumed to be noisy, with different runs of the algorithm possibly producing very different models. We explore to what extent the results just presented were affected by noise in the training procedure. We perform multiple training runs using select evaluation metrics and examining how consistent the resulting models are. This also allows us to determine whether the metric used as a learning criteria influences the stability of learning. For these experiments, Chinese to English models are trained 5 times using a different series of random starting points. As before, 20 random restarts were used during each MERT iteration.

In table 4, models trained to BLEU and METEOR are relatively stable, with the METEOR:0.5 trained models being the most stable. The edit distance models, WER and TERp, vary more across training runs, but still do not exceed the interesting cross metric differences seen in table 1. The instability of WER and TERp, with TERp models having a standard deviation of 1.3 in TERp and 2.5 in BLEU:4, make them risky metrics to use for training.

6 Human Evaluation

The best evaluation metric to use during training is the one that ultimately leads to the best translations according to human judges. We perform a human evaluation of select models using Amazon Mechanical Turk, an online service for cheaply performing simple tasks that require human intelligence. To use the service, tasks are broken down into individual units of work known as human intelligence tasks (HITs). HITs are assigned a small amount of money that is paid out to the workers that complete them. For many natural language annotation tasks, including machine translation evaluation, it is possible to obtain annotations that are as good as those pro-

Train\Eval σ	BLEU:1	BLEU:3	BLEU:4	BLEU:5	TERp	WER	METEOR	METEOR:0.5
BLEU:1	**0.17**	0.56	0.59	0.59	0.36	0.58	0.42	0.24
BLEU:3	0.38	0.41	0.38	0.32	0.70	0.49	0.44	0.33
BLEU:4	0.27	0.29	0.29	0.27	0.67	0.50	0.41	0.29
BLEU:5	**0.17**	0.14	0.19	0.21	0.67	0.75	0.34	0.17
TERp	1.38	2.66	2.53	2.20	1.31	1.39	1.95	1.82
WER	0.62	1.37	1.37	1.25	1.31	1.21	1.10	1.01
METEOR	0.80	0.56	0.48	0.44	3.71	2.69	0.69	1.10
METEOR:0.5	0.32	**0.11**	**0.09**	**0.11**	**0.23**	**0.12**	**0.07**	**0.11**

Table 4: MERT model variation for Chinese to English. We train five models to each metric listed above. The collection of models trained to a given metric is then evaluated using the other metrics. We report the resulting standard devation for the collection on each of the metrics. The collection with the lowest varience is bolded.

Model Pair	% Preferred	p-value
Chinese		
METR R vs. TERp	60.0	0.0028
BLEU:4 vs. TERp	57.5	0.02
NIST vs. TERp	55.0	0.089
NIST vs. TERpA	55.0	0.089
BLEU:4 vs. TERpA	54.5	0.11
BLEU:4 vs. METR R	54.5	0.11
METR:0.5 vs. METR	54.5	0.11
METR:0.5 vs. METR R	53.0	0.22
METR vs. BLEU:4	52.5	0.26
BLEU:4 vs. METR:0.5	52.5	0.26
METR vs. TERp	52.0	0.31
NIST vs. BLEU:4	52.0	0.31
BLEU:4 vs. METR R:0.5	51.5	0.36
WER vs. TERp	51.5	0.36
TERpA vs. TERp	50.5	0.47
Arabic		
BLEU:4 vs. METR R	62.0	< 0.001
NIST vs. TERp	56.0	0.052
BLEU:4 vs. METR:0.5	55.5	0.069
BLEU:4 vs. METR	54.5	0.11
METR R:0.5 vs METR R	54.0	0.14
NIST vs. BLEU:4	51.5	0.36
WER vs. TERp	51.5	0.36
METR:0.5 vs METR	51.5	0.36
TERp vs. BLEU:4	51.0	0.42
BLEU:4 vs. METR R:0.5	50.5	0.47

Table 5: Select pairwise preference for models trained to different evaluation metrics. For A vs. B, *preferred* indicates how often A was preferred to B. We bold the better training metric for statistically significant differences.

duced by experts by having multiple workers complete each HIT and then combining their answers (Snow et al., 2008; Callison-Burch, 2009).

We perform a pairwise comparison of the translations produced for the first 200 sentences of our Chinese to English test data (MT03) and our Arabic to English test data (dev07). The HITs consist of a pair of machine translated sentences and a single human generated reference translation. The reference is chosen at random from those available for each sentence. Capitalization of the translated sentences is restored using an HMM based truecaser (Lita et al., 2003). Turkers are instructed to "...select the machine translation generated sentence that is easiest to read and best conveys what is stated in the reference". Differences between the two machine translations are emphasized by being underlined and bold faced.[9] The resulting HITs are made available only to workers in the United States, as pilot experiments indicated this results in more consistent preference judgments. Three preference judgments are obtained for each pair of translations and are combined using weighted majority vote.

As shown in table 5, in many cases the quality of the translations produced by models trained to different metrics is remarkably similar. Training to the simpler edit distance metric WER produces translations that are as good as those from models tuned to the similar but more advanced TERp metric that allows for swaps. Similarly, training to TERpA, which makes use of both a paraphrase table and edit costs

[9] We emphasize relative differences between the two translations rather than the difference between each translation and the reference in order to avoid biasing evaluations toward edit distance metrics.

tuned to human judgments, is no better than TERp.

For the Chinese to English results, there is a statistically significant human preference for translations that are produced by training to BLEU:4 and a marginally significant preferences for training to NIST over the default configuration of TERp. This contrasts sharply with earlier work showing that TER and TERp correlate better with human judgements than BLEU (Snover et al., 2009; Przybocki et al., 2008; Snover et al., 2006). While it is assumed that, by using MERT, "improved evaluation measures lead directly to improved machine translation quality" (Och, 2003), these results show improved correlations with human judgments are **not always** sufficient to establish that tuning to a metric will result in higher quality translations. In the Arabic results, we see a similar pattern where NIST is preferred to TERp, again with marginal signficance. Strangely, however, there is no real difference between TERp vs. BLEU:4.

For Arabic, training to ranking METEOR is worse than BLEU:4, with the differences being very significant. The Arabic results also trend toward suggesting that BLEU:4 is better than either standard METEOR and METEOR α 0.5. However, for the Chinese models, training to standard METEOR and METEOR α 0.5 is about as good as training to BLEU:4. In both the Chinese and Arabic results, the METEOR α 0.5 models are at least as good as those trained to standard METEOR and METEOR ranking. In contrast to the cross evaluation metric results, where the differences between the α 0.5 models and the standard METEOR models were always fairly dramatic, the human preferences suggest there is often not much of a difference in the true quality of the translations produced by these models.

7 Conclusion

Training to different evaluation metrics follows the expected pattern whereby models perform best on the same type of metric used to train them. However, models trained using the n-gram based metrics, BLEU and NIST, are more robust to being evaluated using the other metrics.

Edit distance models tend to do poorly when evaluated on other metrics, as do models trained using METEOR. However, training models to METEOR can be made more robust by setting α to 0.5, which balances the importance the metric assigns to precision and recall.

The fact that the WER, TER and TERp models perform very similarly suggests that current phrase-based translation systems lack either the features or the model structure to take advantage of swap edit operations. The situation might be improved by using a model that does a better job of both capturing the structure of the source and target sentences and their allowable reorderings, such as a syntactic tree-to-string system that uses contextually rich rewrite rules (Galley et al., 2006), or by making use of larger more fine grained feature sets (Chiang et al., 2009) that allow for better discrimination between hypotheses.

Human results indicate that edit distance trained models such as WER and TERp tend to produce lower quality translations than BLEU or NIST trained models. Tuning to METEOR works reasonably well for Chinese, but is not a good choice for Arabic. We suspect that the newer RYPT metric (Zaidan and Callison-Burch, 2009), which directly makes use of human adequacy judgements of substrings, would obtain better human results than the automated metrics presented here. However, like other metrics, we expect performance gains still will be sensitive to how the mechanics of the metric interact with the structure and feature set of the decoding model being used.

BLEU and NIST's strong showing in both the machine and human evaluation results indicates that they are still the best general choice for training model parameters. We emphasize that improved metric correlations with human judgments do not imply that models trained to a metric will result in higher quality translations. We hope future work on developing new evaluation metrics will explicitly explore the translation quality of models trained to them.

Acknowledgements

The authors thank Alon Lavie for suggesting setting α to 0.5 when training to METEOR. This work was supported by the Defense Advanced Research Projects Agency through IBM. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

References

Abhaya Agarwal and Alon Lavie. 2008. METEOR, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output. In *StatMT workshop at ACL*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *EACL*.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *EMNLP*.

Daniel Cer, Michel Galley, Christopher D. Manning, and Dan Jurafsky. 2010. Phrasal: A statistical machine translation toolkit for exploring new model features. In *NAACL*.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *StatMT workshop at ACL*.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *EMNLP*.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL*.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *JMLR*, 3:951–991.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL*.

Michel Galley, Spence Green, Daniel Cer, Pi-Chuan Chang, and Christopher D. Manning. 2009. Stanford university's arabic-to-english statistical machine translation system for the 2009 NIST evaluation. In *NIST Open Machine Translation Evaluation Meeting*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Alon Lavie and Michael J. Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL*.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *ACL*.

Bonnie Dorr Matthew Snover, Nitin Madnani and Richard Schwartz. 2008. TERp system description. In *MetricsMATR workshop at AMTA*.

Sonja Nießen, Franz Josef Och, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *LREC*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

M. Przybocki, K. Peterson, and S. Bronsart. 2008. Official results of the "Metrics for MAchine TRanslation" Challenge (MetricsMATR08). Technical report, NIST, `http://nist.gov/speech/tests/metricsmatr/2008/results/`.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In *StatMT workshop at EACL)*.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *EMNLP*.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *ICSLP*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher D. Manning. 2005. A conditional random field word segmenter. In *SIGHAN*.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *EMNLP-CoNLL*.

Omar F. Zaidan and Chris Callison-Burch. 2009. Feasibility of human-in-the-loop minimum error rate training. In *EMNLP*, pages 52–61, August.